Process Monitor を用いた RAT 型マルウェアの動的解析

山下恭平、塚本覇虎、奥若菜

2022 年 6 月 14 日

1　テーマ

プロセスのログ解析によるマルウェアの検出を目標に、RAT 型擬似マルウェアツール ShinoBot と監視ツール Process Monitor を用いたマルウェアの動的解析を行う。

2　環境

本実験は以下の県境において実験を行う。

- Virtual Box ・・・ 仮想環境、今回は Windows10 をインストールしている。
- Prosess Monitor ・・・ 動作しているプロセスを監視し、そのログを採取するソフト。
- ShinoBot ・・・ RAT(Random Access Tool) 型のマルウェアを再現したソフト、web 上に設置されたサーバから感染 PC を操作できる。

3　実験

3.1　実験の概要

ShinoBot から感染 PC に命令を送り、特定のァイルを削除した時のプロセスのログと、コマンドプロンプトから直接ファイルを削除したときのプロセスのログを取り、双方のログを比較しながら、RAT 型マルウェアに見られる振る舞いを調査した。

3.2　プロセスの比較

図 1、2 にそれぞれ、ShinoBOT を用いてファイル削除を行なったときと、通常操作でファイルを削除したときの、プロセスの親子関係と採取されたログの量を示した。ShinoBOT を用いる場合、ShinoBOT の実行ファイルが起動している PC に遠隔から命令を送ると、ShinoBOT がコマンドプロンプト (cmd) を呼び出す。次に、コマンドプロンプトは Conhost を呼び出す。Conhost は外部からコマンドプロンプトを実行する際に用いられる実行ファイルである。その後、コマンドプロンプトが powershell を呼び出し、powershell が実際にファイル削除を行う。一方、通常操作の場合、ユーザーがコマンドプロンプトを起動させ、命令コマンドを

直接打ち込まれる。その際、コマンドプロンプトによってまず conhost が呼び出されるが、conhost は何の動作も行わなかった。その後は、ShinoBot の場合と同じく pewershell が起動され、ファイルが削除された。pewershell のログの量や内容から、どちらの場合も削除自体のプロセスにほとんど違いがないことがわかった。

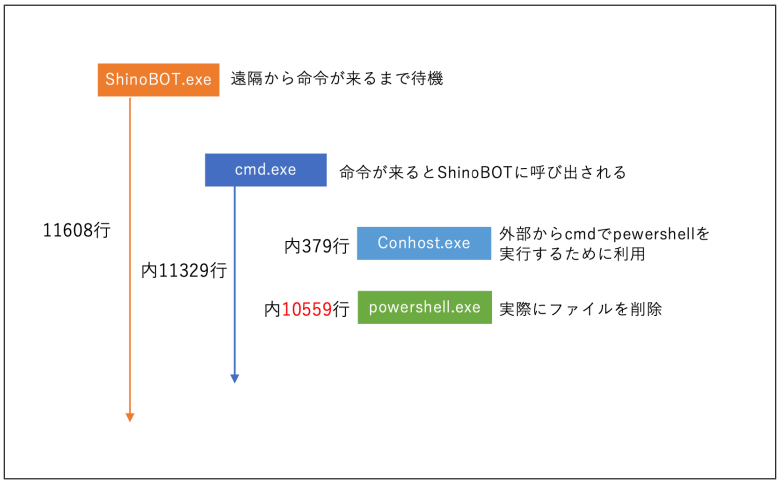


図 1

図 2

3.3 用いられた API の比較

比較を行ったそれぞれのプロセスにおいて、実際にファイルの削除を行なっていた API を確認したところ、同一の API が利用されていることが確認できた。以下はその API の詳細を示したものである。

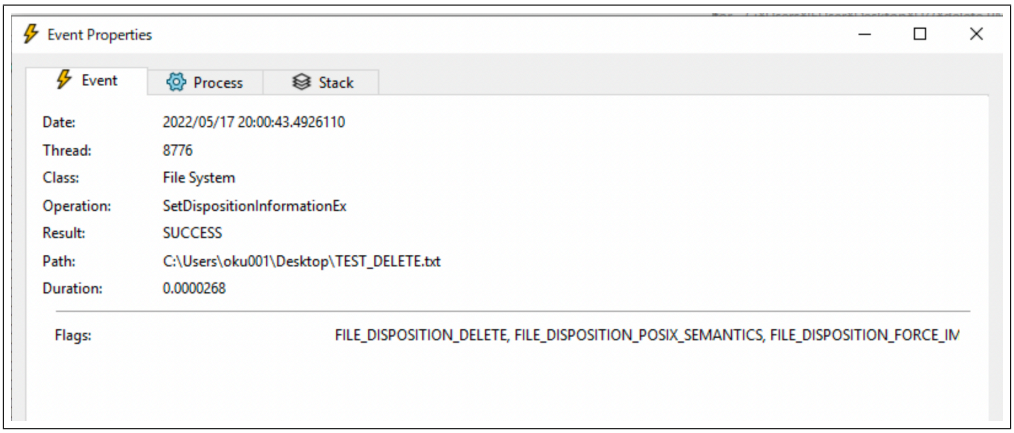


図 3

この API はマイクロソフトの公式ドキュメントにて、内容が公開されていたので、その内容を記載する。(1)

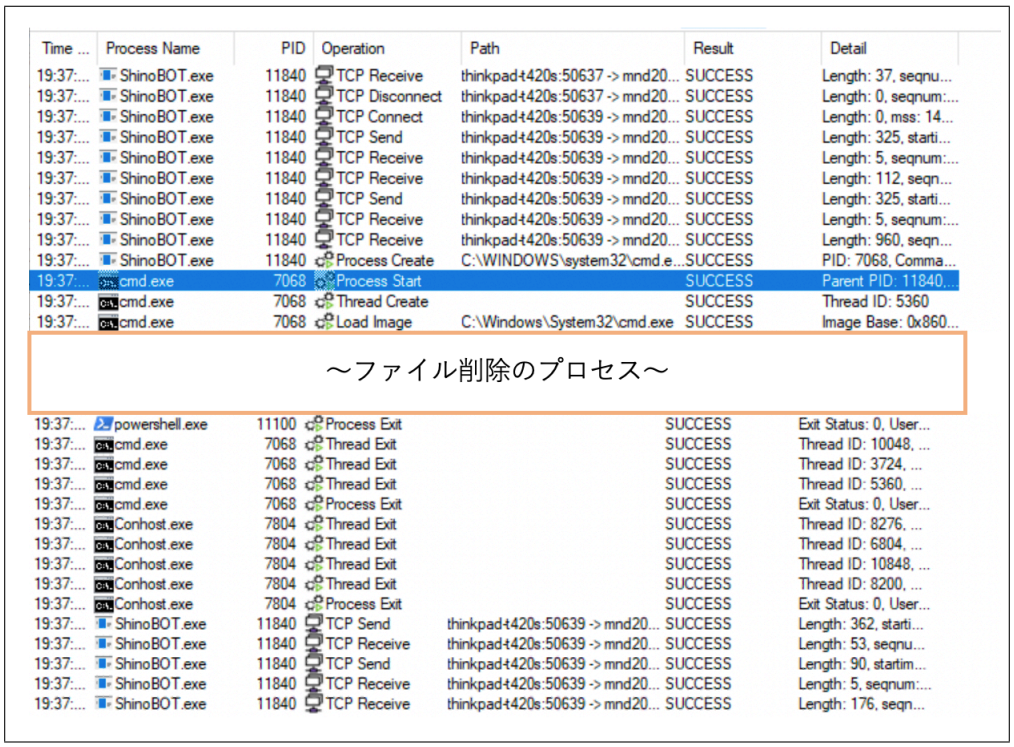
Listing 1　C++

```
1
2  typedef struct _FILE_DISPOSITION_INFORMATION_EX {
3    ULONG Flags;
4  } FILE_DISPOSITION_INFORMATION_EX, *PFILE_DISPOSITION_INFORMATION_EX;
```

フラグ名	意味
FILE_DISPOSITION_DO_NOT_DELETE	システムがファイルを削除しないように指定します。
FILE_DISPOSITION_DELETE	システムがファイルを削除することを指定します。
FILE_DISPOSITION_POSIX_SEMANTICS	システムが POSIX スタイルの削除を実行する必要があることを指定します。
FILE_DISPOSITION_FORCE_IMAGE_SECTION_CHECK	システムがイメージセクションのチェックを強制的に実行するように指定します。
FILE_DISPOSITION_ON_CLOSE	システムが終了時の状態を設定またはクリアするかどうかを指定します。
FILE_DISPOSITION_IGNORE_READONLY_ATTRIBUTE	読み取り専用ファイルの削除を許可します。

表 1

3.4　通信の観測



Time ...	Process Name	PID	Operation	Path	Result	Detail
19:37:...	ShinoBOT.exe	11840	TCP Receive	thinkpad-t420s:50637 -> mnd20...	SUCCESS	Length: 37, seqnu...
19:37:...	ShinoBOT.exe	11840	TCP Disconnect	thinkpad-t420s:50637 -> mnd20...	SUCCESS	Length: 0, seqnum:...
19:37:...	ShinoBOT.exe	11840	TCP Connect	thinkpad-t420s:50639 -> mnd20...	SUCCESS	Length: 0, mss: 14...
19:37:...	ShinoBOT.exe	11840	TCP Send	thinkpad-t420s:50639 -> mnd20...	SUCCESS	Length: 325, starti...
19:37:...	ShinoBOT.exe	11840	TCP Receive	thinkpad-t420s:50639 -> mnd20...	SUCCESS	Length: 5, seqnum:...
19:37:...	ShinoBOT.exe	11840	TCP Receive	thinkpad-t420s:50639 -> mnd20...	SUCCESS	Length: 112, seqn...
19:37:...	ShinoBOT.exe	11840	TCP Send	thinkpad-t420s:50639 -> mnd20...	SUCCESS	Length: 325, starti...
19:37:...	ShinoBOT.exe	11840	TCP Receive	thinkpad-t420s:50639 -> mnd20...	SUCCESS	Length: 5, seqnum:...
19:37:...	ShinoBOT.exe	11840	TCP Receive	thinkpad-t420s:50639 -> mnd20...	SUCCESS	Length: 960, seqn...
19:37:...	ShinoBOT.exe	11840	Process Create	C:\WINDOWS\system32\cmd.e...	SUCCESS	PID: 7068, Comma...
19:37:...	cmd.exe	7068	Process Start		SUCCESS	Parent PID: 11840,...
19:37:...	cmd.exe	7068	Thread Create		SUCCESS	Thread ID: 5360
19:37:...	cmd.exe	7068	Load Image	C:\Windows\System32\cmd.exe	SUCCESS	Image Base: 0x860...
			～ファイル削除のプロセス～			
19:37:...	powershell.exe	11100	Process Exit		SUCCESS	Exit Status: 0, User...
19:37:...	cmd.exe	7068	Thread Exit		SUCCESS	Thread ID: 10048, ...
19:37:...	cmd.exe	7068	Thread Exit		SUCCESS	Thread ID: 3724, ...
19:37:...	cmd.exe	7068	Thread Exit		SUCCESS	Thread ID: 5360, ...
19:37:...	cmd.exe	7068	Process Exit		SUCCESS	Exit Status: 0, User...
19:37:...	Conhost.exe	7804	Thread Exit		SUCCESS	Thread ID: 8276, ...
19:37:...	Conhost.exe	7804	Thread Exit		SUCCESS	Thread ID: 6804, ...
19:37:...	Conhost.exe	7804	Thread Exit		SUCCESS	Thread ID: 10848, ...
19:37:...	Conhost.exe	7804	Thread Exit		SUCCESS	Thread ID: 8200, ...
19:37:...	Conhost.exe	7804	Process Exit		SUCCESS	Exit Status: 0, User...
19:37:...	ShinoBOT.exe	11840	TCP Send	thinkpad-t420s:50639 -> mnd20...	SUCCESS	Length: 362, starti...
19:37:...	ShinoBOT.exe	11840	TCP Receive	thinkpad-t420s:50639 -> mnd20...	SUCCESS	Length: 53, seqnu...
19:37:...	ShinoBOT.exe	11840	TCP Send	thinkpad-t420s:50639 -> mnd20...	SUCCESS	Length: 90, startim...
19:37:...	ShinoBOT.exe	11840	TCP Receive	thinkpad-t420s:50639 -> mnd20...	SUCCESS	Length: 5, seqnum:...
19:37:...	ShinoBOT.exe	11840	TCP Receive	thinkpad-t420s:50639 -> mnd20...	SUCCESS	Length: 176, seqn...

図 4

4　考察

削除におけるプロセスの大部分が一位している点、用いられる API が通常のプロセスと一致している点などを考慮すると。削除を行う一連のプロセス内にマルウェアだと特徴づけるプロセスは存在しないと考えられる。この時、通信ログに着目すると削除のプロセスが生成される前と、一連のプロセスが終了した後に、ShinoBot が TCP 通信を行なっていることが確認できる。RAT に感染しているコンピュータはサーバから送られてくる命令を実行するので、前後の通信によって、命令の受信、命令結果の送信を行なっていると考えらる。RAT に感染したコンピュータから、RAT を検出する方法として、通信ログの確認が有効な手段だと考えられる。